

Projektnavn: **19_cdiodel3**

Gruppe nr: **19**

Afleveringsfrist: **11/5-2015**

Den rapport er afleveret via Campusnet (der skrives ikke under).

Denne rapport indeholder **12** sider inklusiv denne side.

S144847, Nielsen, Jon Tvermose
Kontaktperson(Projektleder)



S125015, Berthold, Ebbe Bjerregaard



S144855, Olsen, Camilla Braae



S144875, Jepsen, Jacob Worck



S144843, Mikkelsen, Christian Flygare Carlend



Indholdsfortegnelse

Indledning	3
Konfiguration.....	3
Opsætning af databasen cdio_db.....	3
Brug af transient datalager	3
Analyse	4
Kravsspecifikation.....	4
Domæne model.....	4
Design	5
Sekvensdiagram	5
Asynkrone kald / Remote Procedure Calls.....	6
Test	8
Unittest	8
Brugertest	8
Login.....	8
Vis personer	8
Tilføj person.....	9
Ret person	9
Slet person	10
Vis recepter	11
Vis PB_info	11
Konklusion.....	12
Fremtidigt arbejde	12

Indledning

Denne rapport er udarbejdet af projektgruppe 19 og indgår som besvarelse af CDIO-opgave til kursus 02324 – Videregående Programmering på Danmarks Tekniske Universitet, forårssemester 2015.

Rapporten dokumenter et system der består af et webinterface med back-end til operatøradministration. Datalaget er implementeret som en database i MySQL, og har også et transient datalager. Der kan skiftes mellem de to datalager ved programmets opstart.

Projektgruppen har igennem længere tid haft et velfungerende samarbejde, hvorfor timeregnskabet er udeladt.

Konfiguration

Afsnittet beskriver hvad afleveringen indeholder samt hvordan programmet startes og bruges. Der forudsættes at der benyttes Eclipse Luna, og at der er installeret MySQL (5.5.42) samt GWT 2.6.1 på computeren.

Afleveringen består af filen 19_cdiodel3.zip. Filen indeholder et projekt som kan importeres og køres i Eclipse. Herudover indeholder filen følgende:

- **19_rapport.pdf** : Denne rapport som dokumenterer systemet
- **cdio_db.sql** : En række SQL statements der opretter de nødvendige tabeller og data for at afvikle programmet med databasen som datalaget

Opsætning af databasen cdio_db

For at køre programmet kræves at data fra cdio_db.sql afvikles på en MySQL database (vrs. 5.5.42) med navn *cdio_db*. Ydermere skal der være adgang med brugernavn '*root*' og intet password.

Databasen skal køres på localhost. Alternativt kan denne opsætning ændres i Eclipse-projektet:

dtu.server.dal.dbconnection.connector.Constant.java

```
mysql> SOURCE C:/Users/JonTvermoseNielsen/Desktop/cdio_db.sql
```

For at indlæse cdio_db.sql kan kommandoen "*SOURCE [filpath]+cdio_db.sql*" benyttes, når man har åbnet sin lokale database (vh.a. "*USE cdio_db*").

Brug af transient datalager

Ønskes programmet testet uden brug af databasen, kan dette gøres med et transient datalag. Til dette formål er klassen *dtu.server.dal.PersonDAOtransient.java* oprettet. For at benytte denne klasse er det nødvendigt at åbne filen *war/WEB-INF/web.xml* og rette i linje 12 hvor ".PersonDAO" erstattes med ".PersonDAOtransient". Herefter kan programmet afvikles med det transiente datalag.

Login

Der kan logges på systemet med et bruger-id og password. Bruger-id er i dette tilfælde *operatoer.opr_id* og password fra *operatoer.password* fra MySQL databasen. Det er kun muligt at logge ind med en bruger der har rollen *admin* som findes i *roller.admin* i databasen. I den vedlagte database er lagt en *admin* ind:

- Bruger-id: 1
- Password: 02324it!

Bemærk at programmet er klargjort til at kunne håndtere flere roller end admin (operatør og farmaceut), men at dette pt. er udkommenteret i koden (dtu.client.ui.Login.java).

Analyse

Afsnittet beskriver kort hvordan systemet er blevet udviklet.

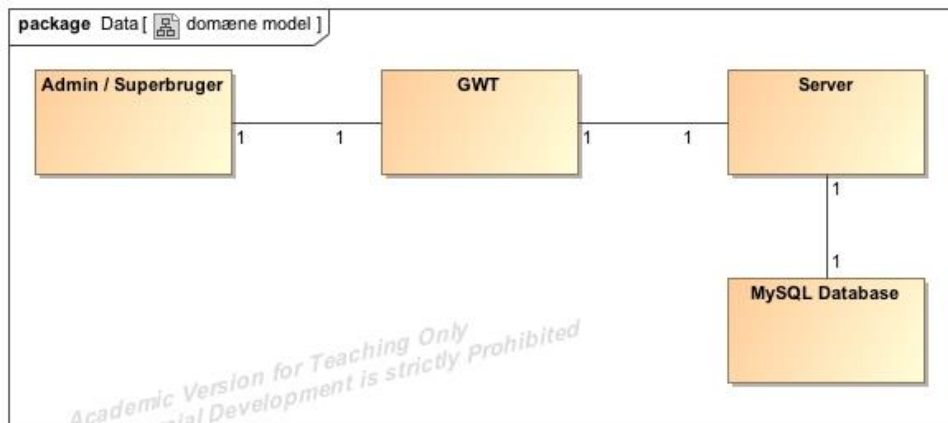
Der er taget udgangspunkt i Kartotek vrs. 2[®] Mads Nyborg fundet på Campusnet.

Kravsspecifikation

Følgende overordnede krav er identificeret:

- Der skal laves et web-interface vha. GWT med back-end til operatøradministration. Følgende funktioner skal være tilgængelige:
 - Opret operatør
 - Opdater operatør
 - Slet operatør
 - Vis operatør(ere)
- Der ønskes lavet to forskellige datalagere:
 - Et transient lager ved anvendelse af passende container-klasser (f.eks. `ArrayList<OperatoerDTO>`)
 - En database i MySQL
- Der skal udarbejdes en lille rapport som dokumenter det samlede system vha. UML.

Domæne model

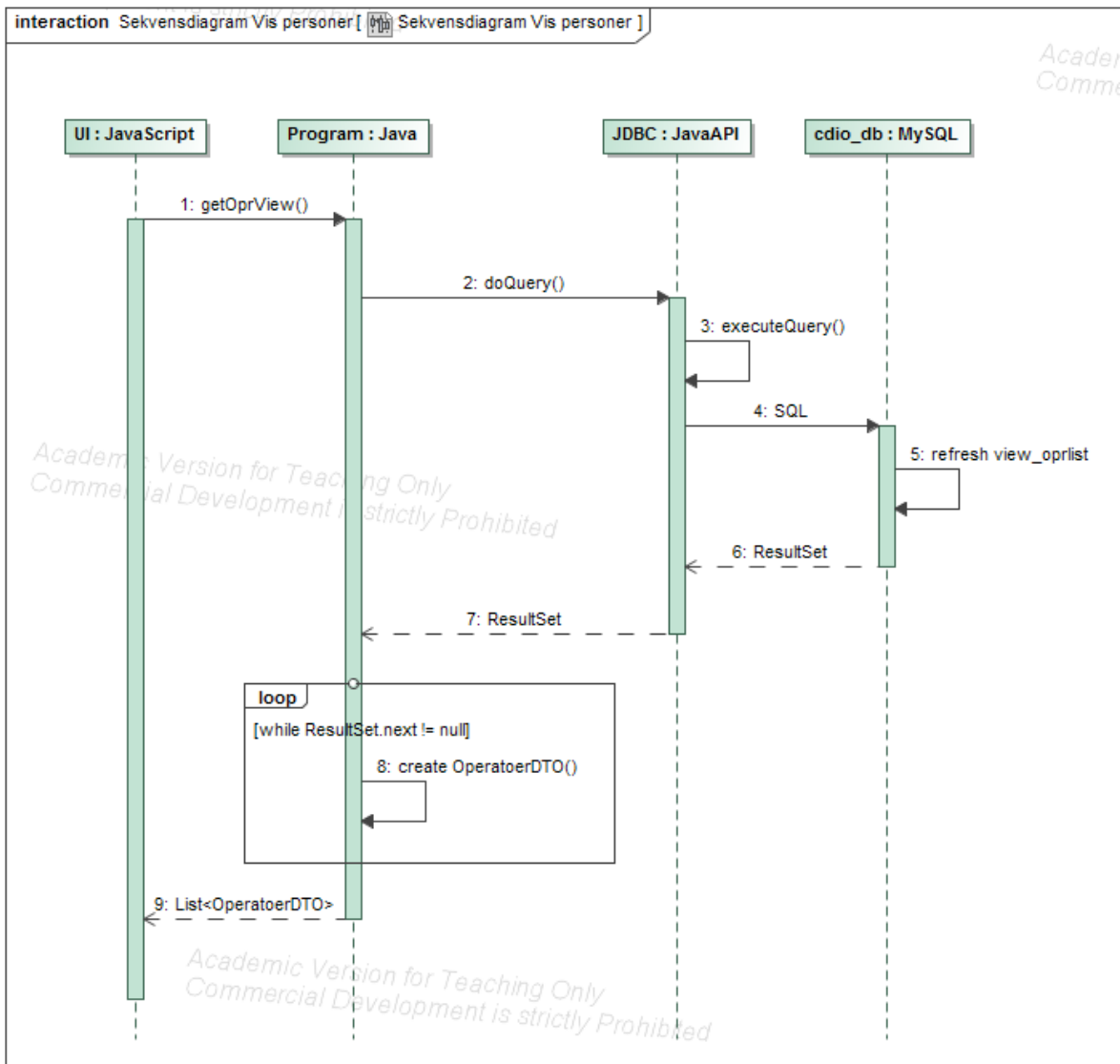


Domæne modellen afspejler de virkelige elementer i systemet. I denne opgave er der lavet et web-interface til en superbruger/admin, som kan tilføje, slette eller rette andre operatørers informationer. For at opfylde dette benyttes Google Web Toolkit, som kan oversætte Java koden til JavaScript på clientsiden. På serveren kører Java koden som ligger i pakken "server.dal". Data lagres i en MySQL database, som serveren tilgår.

Design

Afsnittet dokumenterer hvordan dele af systemet er designet.

Sekvensdiagram



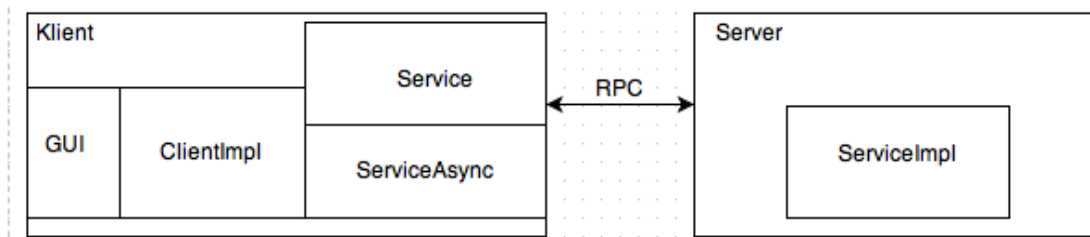
Figuren ovenfor illustrerer et forsimplet sekvensdiagram af funktionen "Vis personer". Det ses at klienten sender besked til serveren om hvilken service der ønskes benyttet. Der sendes samtidig besked om hvilken type resultat der ønskes retur – i dette tilfælde er det udeladt af sekvensdiagrammet. Java-programmet der kører på serveren benytter JDBC og sender en String til dette. JDBC-laget udfører derefter SQL-kommandoen fra den modtagne String på databasen, og modtager et ResultSet retur. Dette returneres til Java-programmet, hvor det behandles og oversættes til Data Transfer Objects (DTO). Herefter laver serveren et Asynkronisk kald til klienten, som modtager en liste af DTO'er. Til slut sørger klienten for at vise resultatet på brugerens skærm.

Asynkrone kald / Remote Procedure Calls

RPC, eller remote procedure calls, benyttes imellem koden på klientsiden og serversiden. Overordnet set kan et RPC forstås som et metodekald til serveren.

Normaltvis når en metode kaldes, gøres dette synkront. Det vil sige at der laves en forespørgsel, og der afventes indtil man modtager svar på forespørgslen.

Den kode der afvikles i browseren er single-trådet, hvilket betyder at ens browser vil hænge, hvis man udfører RPC synkront, idet den vil stå og afvente svar fra serveren. Løsningen på dette er at gøre kaldene asynkrone i stedet, hvilket betyder at browseren ikke hænger, mens den afventer svar fra serveren, når en metode kaldes. I stedet kaldes metoden blot, og når serveren har udført den påkaldte metode, sendes svaret retur til klientsiden.



For at dette kan udføres, deklareres et RPC interface. Dette har essentielt tre komponenter:

- Et serviceInterface (i vores tilfælde KartotekService) som extender RemoteService
- En implementationsklasse (i vores tilfælde KartotekServiceClientImpl) som implementerer det ovenstående interface
- Et asynkront interface (i vores tilfælde KartotekServiceAsync), indeholdende de samme metoder som serviceInterface, som kaldes fra koden på klientsiden.

For at kunne benytte de asynkrone kald, skal disse inddrages i metoderne. Et eksempel på dette ser ud på følgende måde:

```
void savePerson(OperatoerDTO p, AsyncCallback<Void> callback);
```

Som det fremgår af kodeudsnittet, fra KartotekServiceAsync, indgår AsyncCallback altså som parameter i metoden. Idet metodekaldet er asynkront, og det altså ikke i første omgang returnerer noget, er denne form for metoder altid deklareret med void.

```
clientImpl.service.savePerson(newPerson, new AsyncCallback<Void>() {  
  
    @Override  
    public void onSuccess(Void result) {  
        Window.alert("Person gemt i kartotek");  
    }  
  
    @Override  
    public void onFailure(Throwable caught) {  
        Window.alert("Server fejl!" + caught.getMessage());  
    }  
})
```

Til gengæld erklærer man, ved hjælp af `onSuccess` og `onFailure` metoderne, som ses ovenfor, hvordan resultatet skal behandles, når der kommer et svar fra serveren. I dette tilfælde bliver brugeren promptet med de nedenstående udsagn.

Test

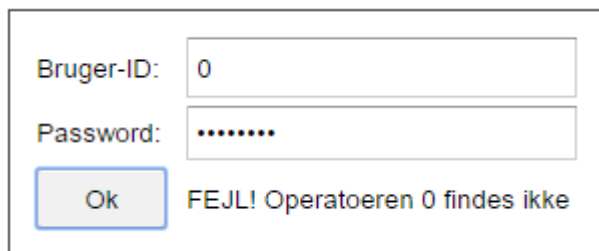
Unittest

Der er lavet en række Unittest som tester serverdelen af programmet, mere specifikt de dele der kommunikerer med MySQL-databasen. Disse test bør afvikles på en test-database, men er i projektet her blot afviklet på den reelle database. Efter JUnits er afviklet, bør database-scriptet køres igen, sådan at databasen nulstilles til det oprindelige udgangspunkt.

Brugertest

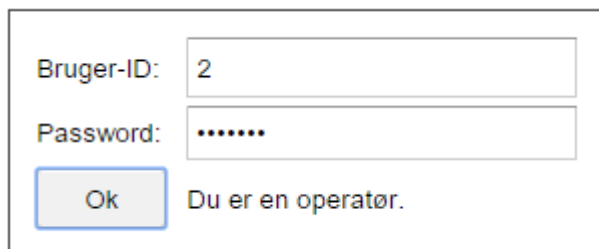
Brugertesten er udført ved at benytte en MySQL database som datalaget.

Login



A login form with two input fields: 'Bruger-ID:' containing '0' and 'Password:' containing seven dots. Below the fields is an 'Ok' button. To the right of the button, a message reads 'FEJL! Operatoeren 0 findes ikke'.

I ovenstående figur forsøges der at logge ind med et bruger-id der ikke findes i databasen. Systemet melder korrekt tilbage at operatøren ikke findes.



A login form with two input fields: 'Bruger-ID:' containing '2' and 'Password:' containing seven dots. Below the fields is an 'Ok' button. To the right of the button, a message reads 'Du er en operatør.'

Herefter forsøges at logge ind med en operatør som ikke har rollen "admin". Der identificeres korrekt at brugeren findes, og at brugeren har rollen "operatoer". Der nægtes login.

Herefter logges der ind med korrekt bruger-id (1) og password (02324it!). Der gives som forventet adgang til systemet.

Vis personer

[Vis personer](#)
[Tilføj person](#)
[Ret person](#)
[Slet person](#)
[Vis recepter](#)
[Vis PB_info](#)
[Log ud](#)

Skjul detaljer

Skjul inaktive

☐ Vis KUN inaktive

Id	Navn	Initial	admin	operatoer	farmaceut
1	Angelo A	AA	X	X	-
2	Antonella B	AB	-	X	-
3	Luigi C	LC	-	X	X

Menupunktet "Vis personer" vælges, og man præsenteres af en oversigt over hvilke personer der findes i databasen og har mindst en rolle. I ovenstående billede er valgt at få vist detaljer om roller, samt at få vist eventuelle inaktive personer (en person uden en rolle). Der kan ligeledes vælges kun at vi vist inaktive personer.

Tilføj person

[Vis personer](#)
[Tilføj person](#)
[Ret person](#)
[Slet person](#)
[Vis recepter](#)
[Vis PB_info](#)
[Log ud](#)

Navn

Anders And

Cpr

123456-7890

Admin

☐

Operator

☒

Farmaceut

☒

Tilføj

Siden på 127.0.0.1:8888 siger:

Person gemt i kartotek

OK

Ovenfor er menupunktet "Tilføj person" valgt, og felter er udfyldt med gyldig data (tjekket sker i java under indtastning/afkrydsning). Herefter er der klikket på knappen "Tilføj". Der modtages besked retur fra serveren (som kommunikerer med databasen), at personen er gemt i databasen.

Personen fremgår herefter på oversigten "Vis personer".

[Vis personer](#)
[Tilføj person](#)
[Ret person](#)
[Slet person](#)
[Vis recepter](#)
[Vis PB_info](#)
[Log ud](#)

Skjul detaljer

Skjul inaktive

☐ Vis KUN inaktive

Id	Navn	Initial	admin	operatoer	farmaceut
1	Angelo A	AA	X	X	-
2	Antonella B	AB	-	X	-
3	Luigi C	LC	-	X	X
4	Anders And	And	-	X	X

Ret person

[Vis personer](#)
[Tilføj person](#)
[Ret person](#)
[Slet person](#)
[Vis recepter](#)
[Vis PB_info](#)
[Log ud](#)

Id	Navn	Initialer	CPR	Admin	Operatoer	Farmaceut	
1	Angelo A	AA	070770-7007	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
2	Antonella B	AB	080880-8008	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	edit
3	Luigi C	LC	090990-9009	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	edit
4	<div>Andersine And</div>	<div>AT</div>	<div>123456-7890</div>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<div>ok cancel</div>

Ovenfor er valgt "Ret person". Navn og initialer ændres på person med Id = 4, og der klikkes på "ok".

Et kig direkte i databasen bekræfter at personen er oprettet, og at ændringen er slået igennem til databasen:

```
mysql> select * from operatoer;
+-----+-----+-----+-----+-----+
| opr_id | opr_navn | ini | cpr | password |
+-----+-----+-----+-----+-----+
| 1 | Angelo A | AA | 070770-7007 | 02324it! |
| 2 | Antonella B | AB | 080880-8008 | atoJ21v |
| 3 | Luigi C | LC | 090990-9009 | jEfm5aQ |
| 4 | Andersine And | AT | 123456-7890 | 02324it! |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Slet person

Vis personer
Tilføj person
Ret person
Slet person
Vis receptor
Vis PB_info
Log ud

Id	Navn	
1	Angelo A	delete
2	Antonella B	delete
3	Luigi C	delete
4	Andersine And	ok cancel

Siden på 127.0.0.1:8888 siger:

Andersine And deaktiveret

OK

Menupunktet Slet personer viser de personer som har mindst en rolle. Der vælges at slette personen med Id = 4, og der returneres svar fra serveren at personen er deaktiveret. Det er værd at bemærke at personen ikke er slettet fra databasen, men at Slet person-funktionen 'blot' sætter en persons roller til at være false (dvs. personen har ingen roller og er dermed inaktiv). Dette gøres for at bibeholde historik i databasen.

Et kig i databasen bekræfter dette:

```
mysql> select * from operatoer
-> natural join roller;
+-----+-----+-----+-----+-----+-----+-----+
| opr_id | opr_navn | ini | cpr | password | admin | operatoer | farmaceut |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Angelo A | AA | 070770-7007 | 02324it! | 1 | 1 | 0 |
| 2 | Antonella B | AB | 080880-8008 | atoJ21v | 0 | 1 | 0 |
| 3 | Luigi C | LC | 090990-9009 | jEfm5aQ | 0 | 1 | 1 |
| 4 | Andersine And | AT | 123456-7890 | 02324it! | 0 | 0 | 0 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Vis recepter

[Vis personer](#)
[Tilføj person](#)
[Ret person](#)
[Slet person](#)
[Vis recepter](#)
[Vis PB_info](#)
[Log ud](#)

Recept id	Recept	Ingredienser	
		Råvare	Vægt
1	margherita		
		dej	10
		tomat	2
		ost	2
			Skjul
2	prosciutto	Vis	
3	capricciosa	Vis	

Vis recepter viser de recepter der findes i databasen, samt hvilke ingredienser de består af (råvarenavn og vægt).

Vis PB_info

[Vis personer](#)
[Tilføj person](#)
[Ret person](#)
[Slet person](#)
[Vis recepter](#)
[Vis PB_info](#)
[Log ud](#)

Recept id	Recept navn	Netto vægt	PB id	Opr id	Opr navn	Status
1	margherita	10.05	1	1	Angelo A	2
1	margherita	2.03	1	1	Angelo A	2
1	margherita	1.98	1	1	Angelo A	2
1	margherita	10.01	2	2	Antonella B	2
1	margherita	1.99	2	2	Antonella B	2
1	margherita	1.47	2	2	Antonella B	2
2	prosciutto	10.07	3	1	Angelo A	2
2	prosciutto	2.06	3	2	Antonella B	2
2	prosciutto	1.55	3	1	Angelo A	2
2	prosciutto	1.53	3	2	Antonella B	2
3	capricciosa	10.02	4	3	Luigi C	1
3	capricciosa	1.57	4	3	Luigi C	1
3	capricciosa	1.03	4	3	Luigi C	1
3	capricciosa	0.99	4	3	Luigi C	1

Vis PB_info viser information om de forskellige produktbatchkomponenter. Her er det muligt at identificere hvilken operatør der har lavet hvilken produktbatchkomponent, samt status på produktionen.

Konklusion

Der er lavet et system som består af en klient, en server samt en database. Alle tre enheder kommunikerer sammen. Der er herudover udviklet et transient datalager som kan benyttes når der arbejdes på klient-delen af projektet.

Alle funktioner er blevet testet vha. en brugertest, og der er herudover lavet en række unit-test der kan afvikles på serverdelen/databasedelen af systemet.

Fremtidigt arbejde

Der er i projektet fokuseret på at implementere funktioner der relaterer sig til operatøradministration, dvs., oprette, ændre, vise og slette (deaktivere) brugere. Der bør oprettes et yderligere antal views, funktioner og procedurer der understøtter operationer for recepter, råvarer og produktbatches.

Ydermere bør der adgangs begrænses til databasen, således at der oprettes adgangsroller til databasen, der komplimenterer de roller en person i systemet kan have. Man kunne forestille sig at f.eks. en operatørrolle kun har adgang til SELECT fra diverse views, hvor en farmaceutrolle kunne have adgang til at tilføje, opdatere og slette recepter via en række procedurer. Ligeledes bør farmaceutrollen ikke have fuld adgang til SELECT på databasen, men blot på en række views.